Optimizing Revenue over Data Driven Assortments

Deeksha Sinha
MIT

Theja Tulabandhula
UIC

jda. | Category Management

NIELSEN
ASSORTMENT &
SPACE OPTIMISATION
FASTEST ▸ MOST ACCURATE ▸ GROWTH

4R

Challenges

- Retailers are dealing with a large selection of products
  - Walmart has ~21,000 (n) SKUs of cookies
  - Typically carries ~500 (C) SKUs in a store
- Combinatorial number of possible assortments: n^C
- E-commerce platforms need to take assortment decision in < 100ms

Assortment optimization (AO) algorithms need to be scalable

Challenges

- Business rules govern feasible assortments
- Constraints based on
  - Assortment size
  - No. of products from different sub-groups
  - Inventory levels
  - Diversity
  - Frequent itemset mining (based on observed sales patterns)

AO algorithm needs to be able to incorporate constraints

Contributions

- Propose an AO algorithm that finds approximately optimal solution over an **arbitrary set** of constraints
  - time **sub-linear** in the no of feasible assortments
- For cardinality constrained AO, find approximately optimal solution in time **linear** in the no. of products
- Algorithm is relevant in both online and offline settings
- Illustrate superior performance over real and synthetic datasets as compared to existing approaches
- Solve problems with large no of products (~20,000)

Assortment Optimization

- n products with prices $p_1 \geq p_2 \cdots \geq p_n$
- Assortment $S \subset \{1, 2, \cdots n\}$
- Set of all feasible assortments $\mathscr{S} \subset 2^{[n]}$ with $|\mathscr{S}| = N$
- Customer preference encoded by v
- Probability that customer purchases product i is $P_v(i|S)$
- Revenue of assortment S is $R_v(S) = \sum_{i \in S} p_i P_v(i|S)$

AO problem: $\max_{S \in \mathscr{S}} \sum_{i \in S} p_i P_v(i|S)$

Customer Choice Model

Multiple choice models have been proposed:

- Multinomial logit (MNL) - McFadden, 1974
- Mixture of MNLs - Boyd and Mellman, 1980 and Cardell and Dunbar, 1980
- Nested logit - Ben-Akiva, 1985
- Multinomial probit - Daganzo, 1979
- Exponomial choice model - Alptekinoğlu and Semple, 2016
- Markov chain choice model - Blanchet et al., 2016
- Distribution over rankings - Farias et al., 2013

Customer Choice Model

- MNL model is the most commonly used choice model
  - Easy estimation of parameters
  - Optimization problem is tractable
- MNL choice model:
  - Parameters $\mathbf{v} = (v_1, v_2, \cdots, v_n)$
  - v_i encodes preference of user for purchasing product i
  - $P_v(i|S) = \frac{v_i}{1 + \sum_{j \in S} v_j}$

Existing Literature

- Unconstrained and cardinality constrained AO are well studied

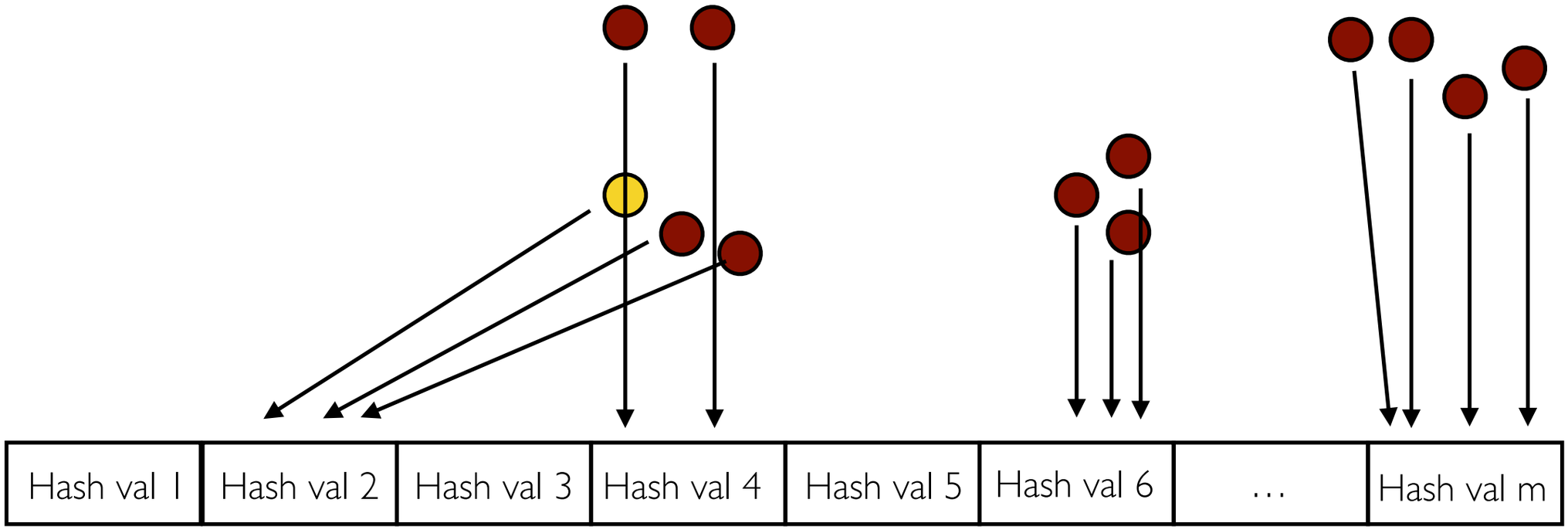
Work	Choice model	Constraints	Approach	Guarantee	# Products
Rusmevichinton g et al., 2010	MNL	Cardinality	Specialized (Static-MNL)	$O(n^2 \log n)$	200
Davis et al., 2013	MNL	Totally unimodular	Linear Programming	$O(n^{3.5})$	-
Jagabathula, 2014	MNL	Cardinality	Greedy (ADXopt)	$O(n^2C^2)$	15
Bertsimas & Misic, 2015	Rankings	Cardinality, precedence	Integer programming	-	40
Desir et al., 2015	Markov chain	Cardinality	Dynamic programming	-	200
Sinha & Tulabandhula, 2017	**MNL**	**Cardinality Arbirary**	**Binary search & Hashing**	$O(n \log C)$ $O(nN^\rho)$	**20000**

Locality Sensitive Hashing

- Find vectors from a set of vectors which are 'similar' to query vector
  - Find *neighbors* of query vector in Euclidean space
- Uses hash functions which produce similar outputs for neighboring vectors. Eg -
  - Vectors x, y with norm 1
  - Sample $a \sim MVN(0, I)$
  - Hash function: $h_a(x) = sign(a \cdot x)$
  - If $x \cdot y \geq D$, then $P_a(h_a(x) = h_a(y)) \geq 1 - \frac{1}{\pi} \cos^{-1} D$
  - Aggregate multiple hash functions to control collision probability

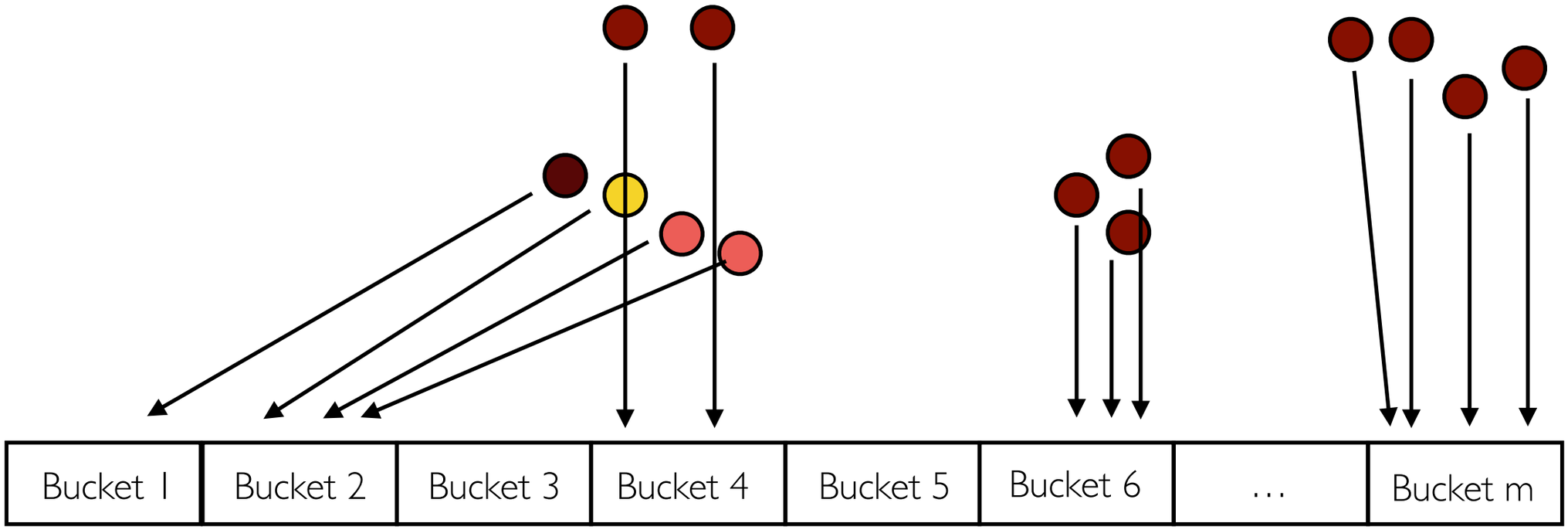
Locality Sensitive Hashing

- Given vectors in search space
- Pre-processed by multiple hash functions
- At run-time, query vector is hashed and the vectors with which the hash collides are retrieved



Locality Sensitive Hashing

- Given vectors in search space
- Pre-processed by multiple hash functions
- At run-time, query vector is hashed and the vectors with which the hash collides are retrieved

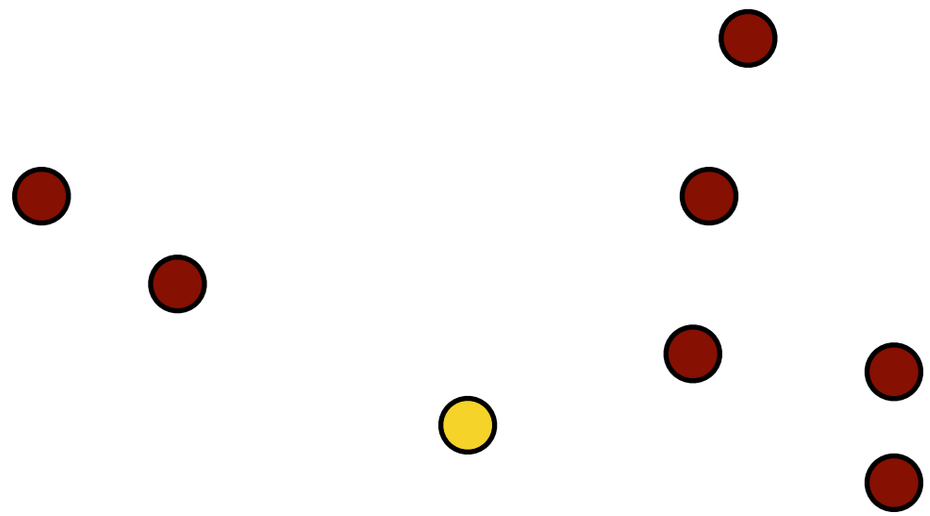


Locality Sensitive Hashing

- Given vectors in search space
- Pre-processed by multiple hash functions
- At run-time, query vector is hashed and the vectors with which the hash collides are retrieved
- Run time operations: hashing query vector and finding vectors with same hash value

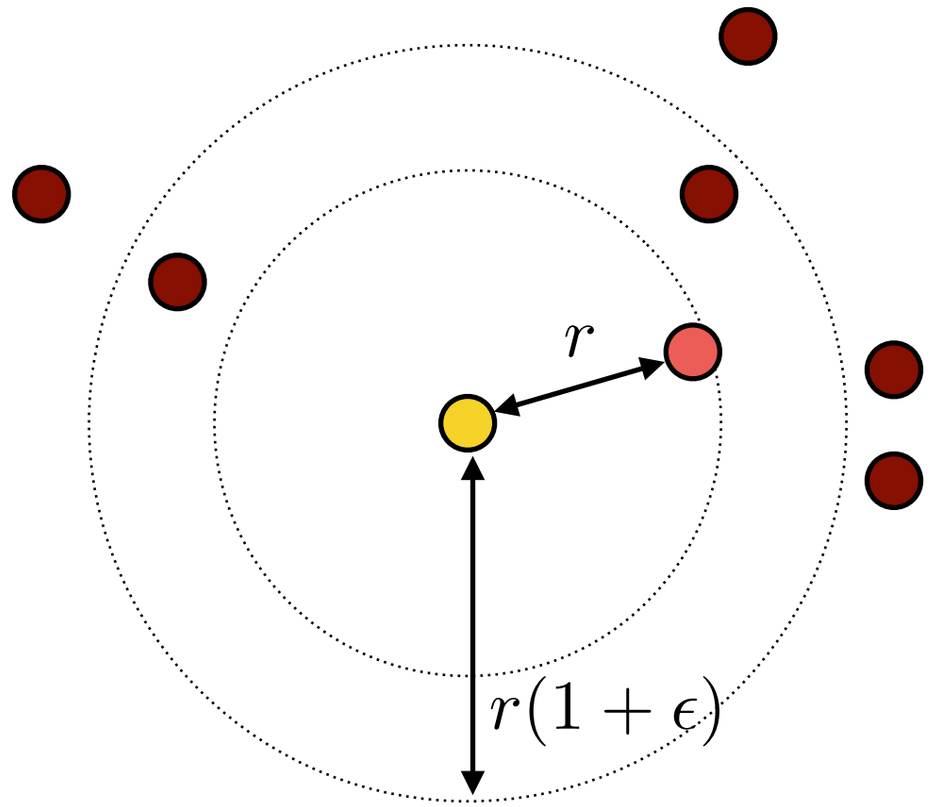
Locality Sensitive Hashing

- LSH calculates 'approximate' nearest neighbors efficiently



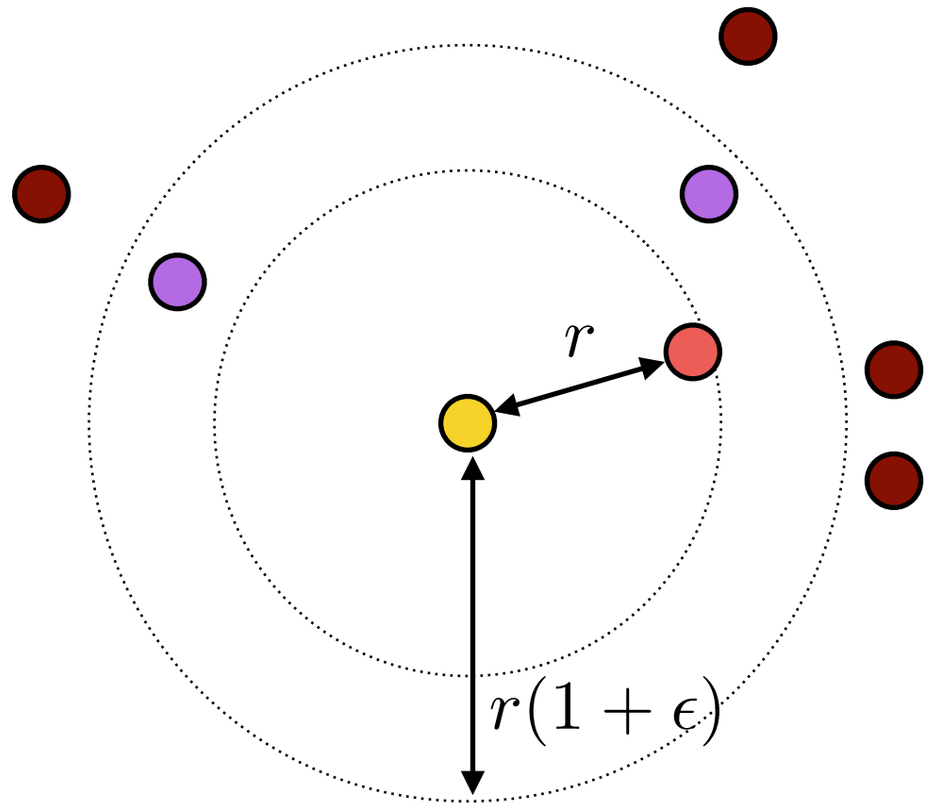
Locality Sensitive Hashing

- LSH calculates 'approximate' nearest neighbors efficiently



Locality Sensitive Hashing

- LSH calculates 'approximate' nearest neighbors efficiently
- For N points in d dimensions (and $\rho < 1$)
  - Time complexity: $O(dN^{\rho})$
  - Space complexity: $O(dN^{1+\rho})$

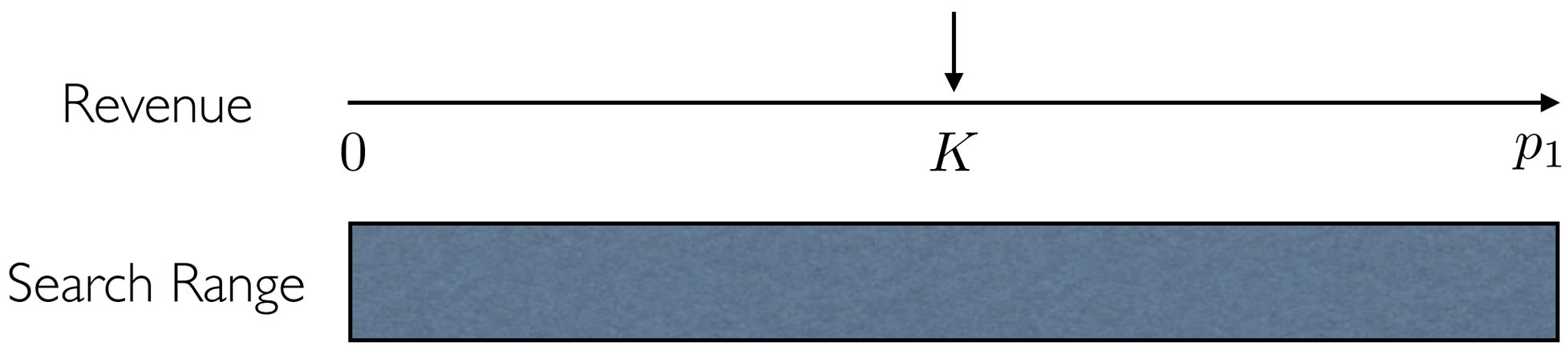


Assort-MNL Algorithm

Binary search based AO algorithm

- Initialize search range: $L = 0, U = p_1$
- Repeat:
  - $K = (U + L)/2$
  - Is there a feasible assortment with revenue $R_v(S) \geq K$?

    Yes: Update $L = K$



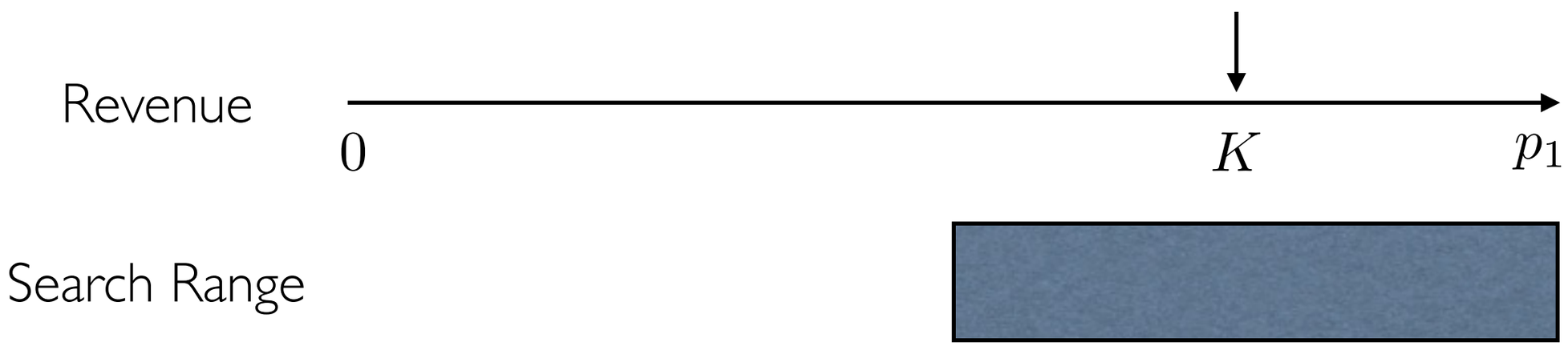
Assort-MNL Algorithm

Binary search based AO algorithm

- Initialize search range: $L = 0, U = p_1$
- Repeat:
  - $K = (U + L)/2$
  - Is there a feasible assortment with revenue $R_v(S) \geq K$?

    Yes: Update $L = K$

    No: Update $U = K$



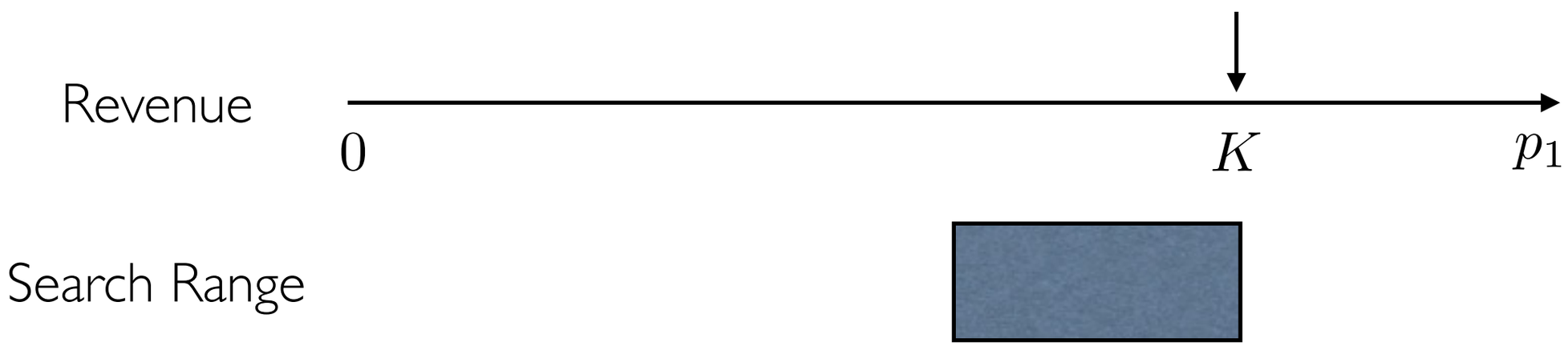
Assort-MNL Algorithm

Binary search based AO algorithm

- Initialize search range: $L = 0, U = p_1$
- Repeat:
  - $K = (U + L)/2$
  - Is there a feasible assortment with revenue $R_v(S) \geq K$?

    Yes: Update $L = K$

    No: Update $U = K$



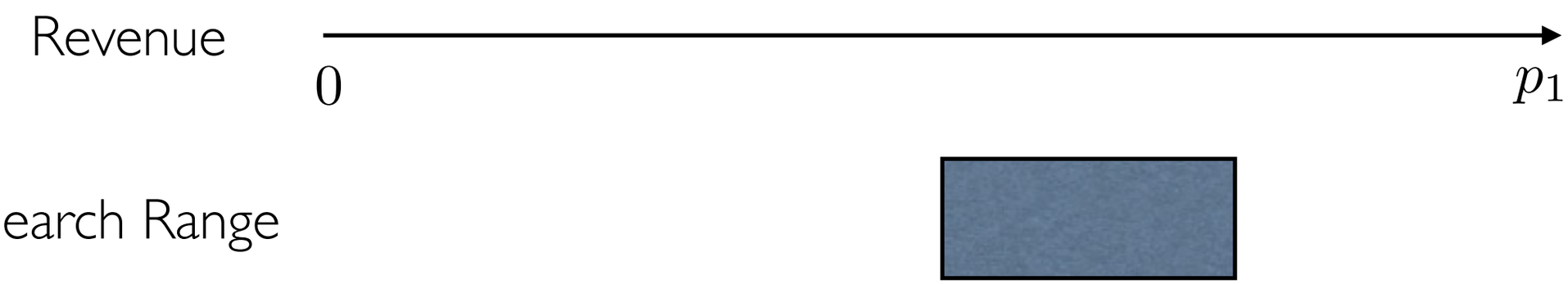
Assort-MNL Algorithm

Binary search based AO algorithm

- Initialize search range: $L = 0, U = p_1$
- Repeat till search range is narrowed to desired tolerance level:
  - $K = (U + L)/2$
  - Is there a feasible assortment with revenue $R_v(S) \geq K$?

    Yes: Update $L = K$

    No: Update $U = K$



Assort-MNL Algorithm

Binary search based AO algorithm

- Initialize search range: $L = 0, U = p_1$
- Repeat till search range is narrowed to desired tolerance level:
  - $K = (U + L)/2$
  - Is there a feasible assortment with revenue $R_v(S) \geq K$?
    Yes: Update $L = K$
    No: Update $U = K$

Obtain approximately optimal revenue i.e. revenue within tolerance of the optimal revenue

Compare Step

- Does there exist a feasible assortment with $R_v(S) \geq K$?
  - Find if $\exists\ S \in \mathscr{S}$ s.t. $\dfrac{\sum_{i \in S} p_i v_i}{1 + \sum_{j \in S} v_j} \geq K$

Compare Step

- Does there exist a feasible assortment with $R_v(S) \geq K$?
  - Find if $\exists\ S \in \mathscr{S}$ s.t. $\sum_{i \in S} v_i(p_i - K) \geq K$

Compare Step

- Does there exist a feasible assortment with $R_v(S) \geq K$?
  - Find if $\max_{S \in \mathscr{S}} \sum_{i \in S} v_i(p_i - K) \geq K$

Compare Step

- Does there exist a feasible assortment with $R_v(S) \geq K$?
  - Optimization problem: $\max_{S \in \mathscr{S}} \sum_{i \in S} v_i(p_i - K)$
- Any MIPS sub-routine can be used
- MIPS can be solved approximately using LSH
- Every MIPS query can be solved in $O(nN^{\rho})$
- Time sub-linear in the no of feasible assortments

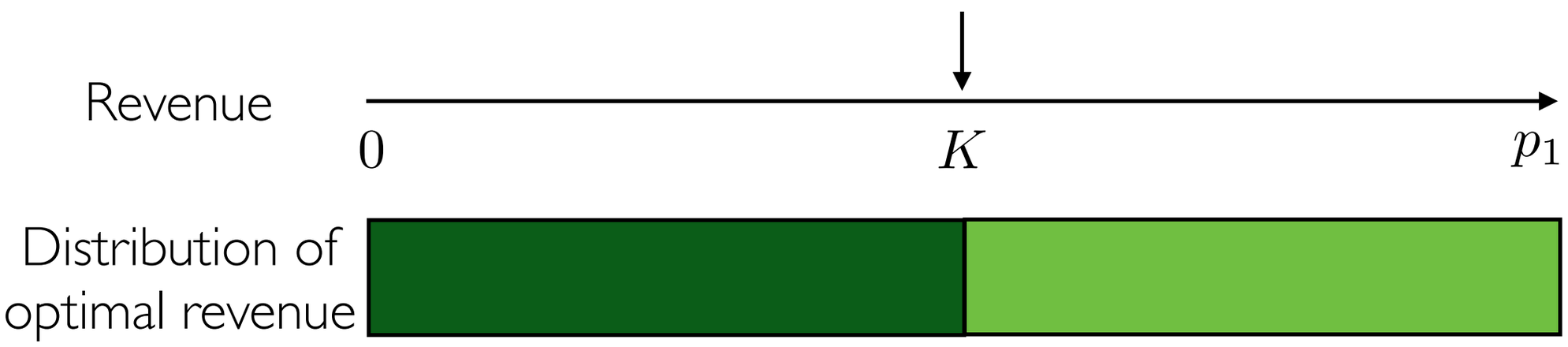
Maximum inner product search problem (MIPS)

Assort-MNL(BZ)

- Approximation error: LSH data structure fails
- Modify binary search to accommodate approximation
- Build on noisy binary search algo (Burnashev and Zigangirov, 1974)
  - Maintain prior distribution over optimal revenue
  - 'Bisect' at the median of prior distribution
  - Update posterior based on comparison

Assort-MNL(BZ)

- Approximation error: LSH data structure fails
- Modify binary search to accommodate approximation
- Build on noisy binary search algo (Burnashev and Zigangirov, 1974)
  - Maintain prior distribution over optimal revenue
  - 'Bisect' at the median of prior distribution
  - Update posterior based on comparison



Assort-MNL(BZ)

- Approximation error: LSH data structure fails
- Modify binary search to accommodate approximation
- Build on noisy binary search algo (Burnashev and Zigangirov, 1974)
  - Maintain prior distribution over optimal revenue
  - 'Bisect' at the median of prior distribution
  - Update posterior based on comparison

Assort-MNL(BZ)

- Approximation error: LSH data structure fails
- Modify binary search to accommodate approximation
- Build on noisy binary search algo (Burnashev and Zigangirov, 1974)
  - Maintain prior distribution over optimal revenue
  - 'Bisect' at the median of prior distribution
  - Update posterior based on comparison

Assort-MNL(BZ): Guarantees

Theorem [S., Tulabandhula]: Under mild assumptions, T iterations of Assort-MNL(BZ) having time complexity $O(nN^{\rho})$ each, lead to

$$P(|R^* - R| > \delta) \leq \frac{p_1 - \delta}{\delta} \left(\sqrt{\alpha} + 0.5\right)^T$$

where R^* is the revenue of the optimal assortment, α is the failure probability of the LSH structure, δ is the tolerance level.

- Error probability decays exponentially
- Time taken is sub-linear in the number of assortments

Assort-MNL: Cardinality constrained

Compare step optimization: $\max_{|S| \leq C} \sum_{i \in S} v_i(p_i - K)$

- Rank products by $v_i(p_i - K)$
- Choose top C products having +ve value of $v_i(p_i - K)$
- Time complexity: $O(n \log C)$
- No of iterations to get $\epsilon-$optimal revenue: $\log\left(\frac{p_1}{\epsilon}\right)$

Theorem [S., Tulabandhula]: Assort-MNL (Card) produces $\epsilon-$optimal revenue in time $O\left(n \log C \log\left(\frac{p_1}{\epsilon}\right)\right)$.

Experiments: Dataset

- Two real data sets:
  - Billion Prices Project (Cavallo 2016): daily prices for all goods sold by 7 large retailers in Latin America and USA between 2007-2010
    - choose 2 retailers with10 M daily observations for 94,000 items and 5 M daily observations for 30,000 items
    - use prices from 50 different days to generate instances for 50 Monte Carlo runs
  - MNL parameters are chosen from $U[0, 1]$

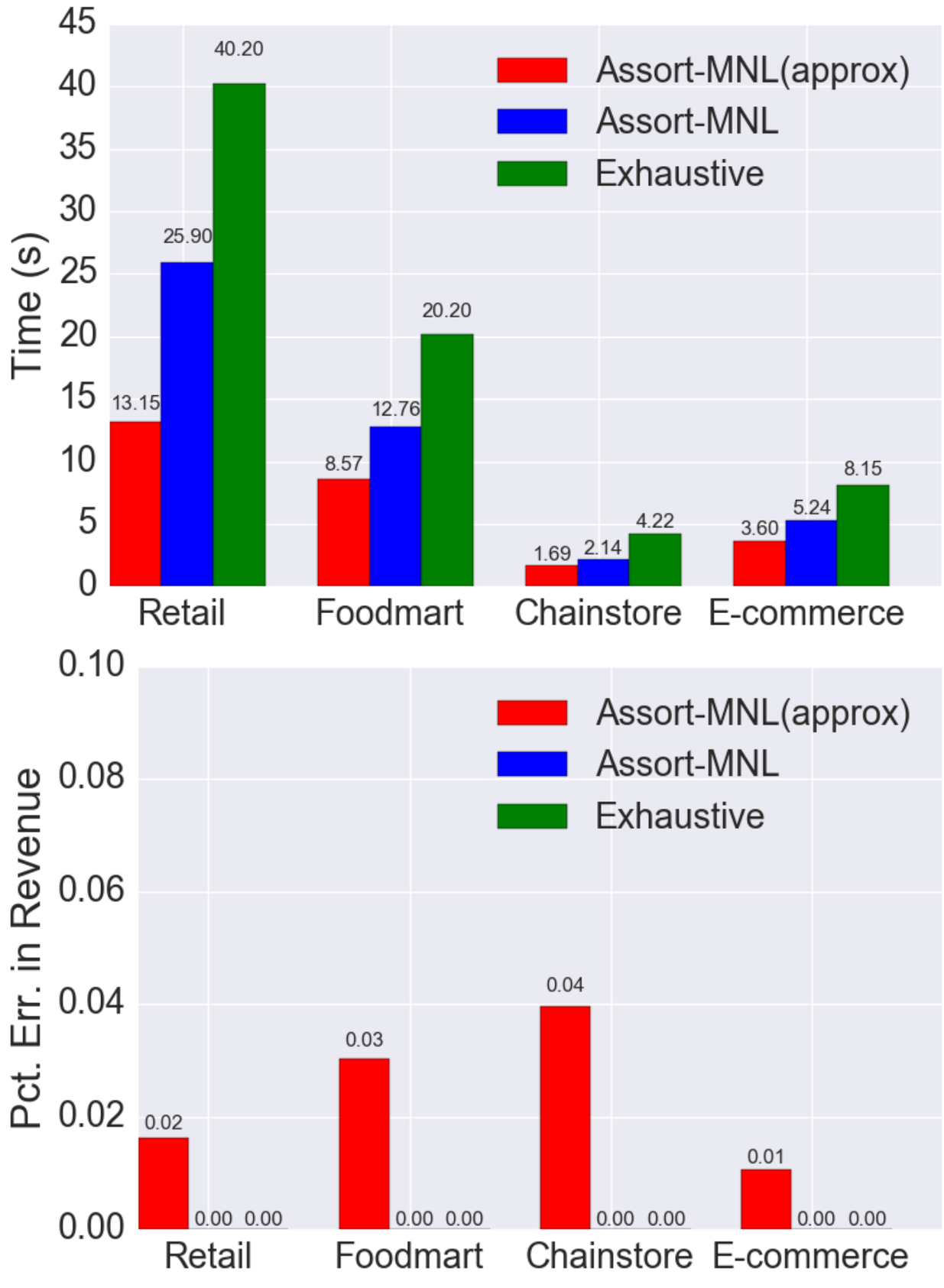
Experiments: Dataset

- Retail, foodmart, chainstore and e-commence transaction logs (Fournier-Viger et al., 2014)
  - generate frequent item sets with apt min. support using FPgrowth algorithm
  - create 50 instances by generating price and MNL parameters from $U[0, 1000]$ and $U[0, 1]$

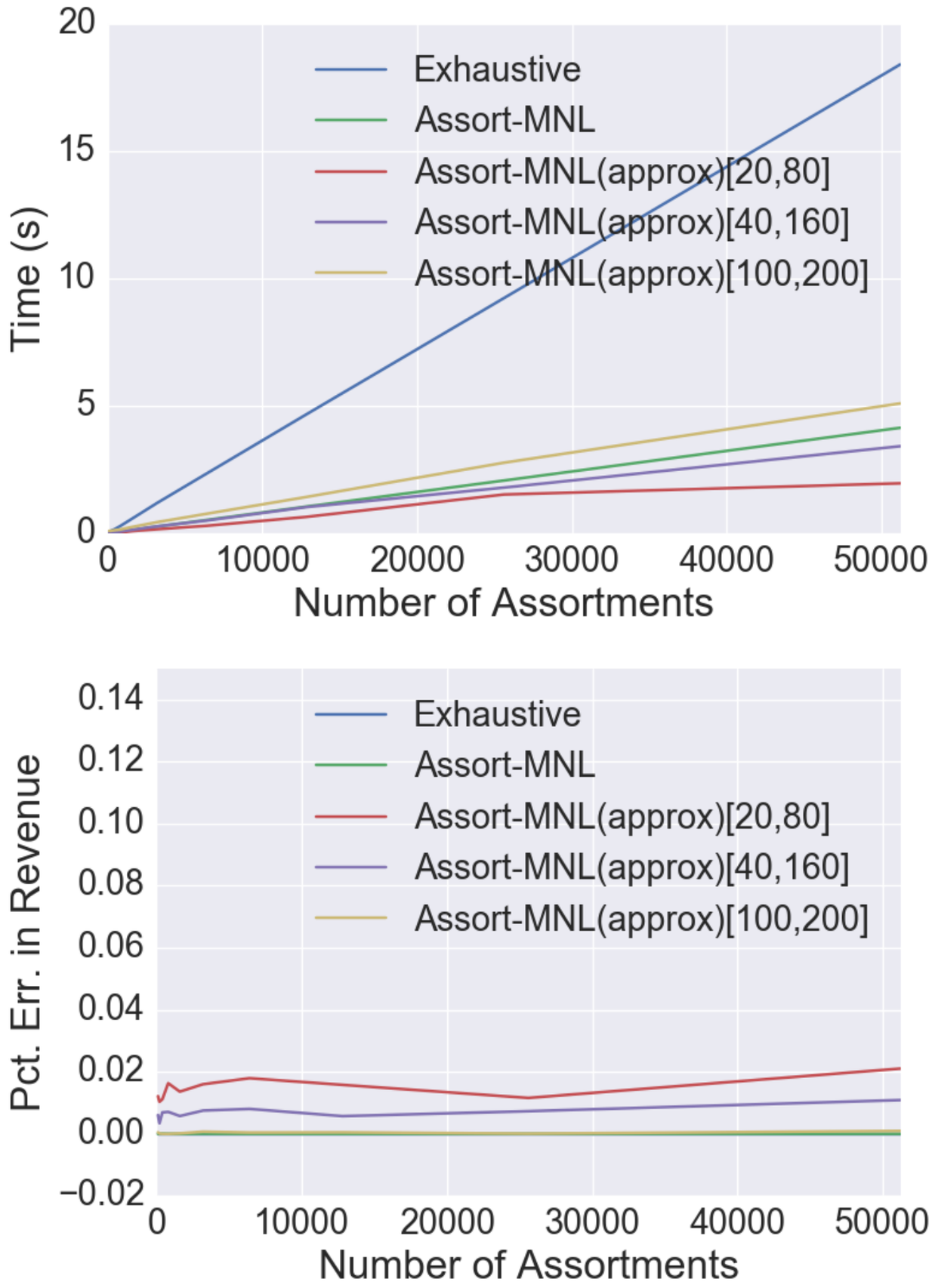
Table 1 Assortments generated from transactional datasets.

Dataset	Retail	Foodmart	Chainstore	E-commerce
Number of transactions	88162	4141	1112949	540455
Number of items	3160	1559	321	2208
Number of general assortments	80524	81274	75853	23276
Size of largest assortment	12	14	16	8
Size of smallest assortment	3	4	5	3

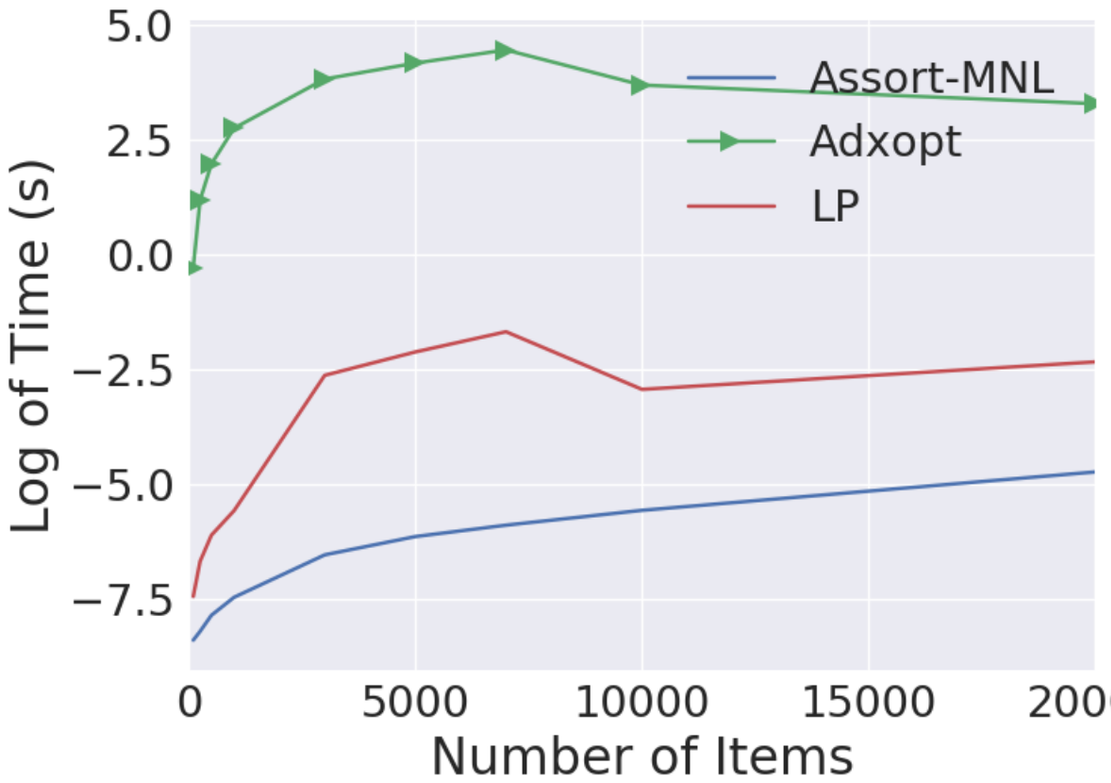
Experiments: General assortments

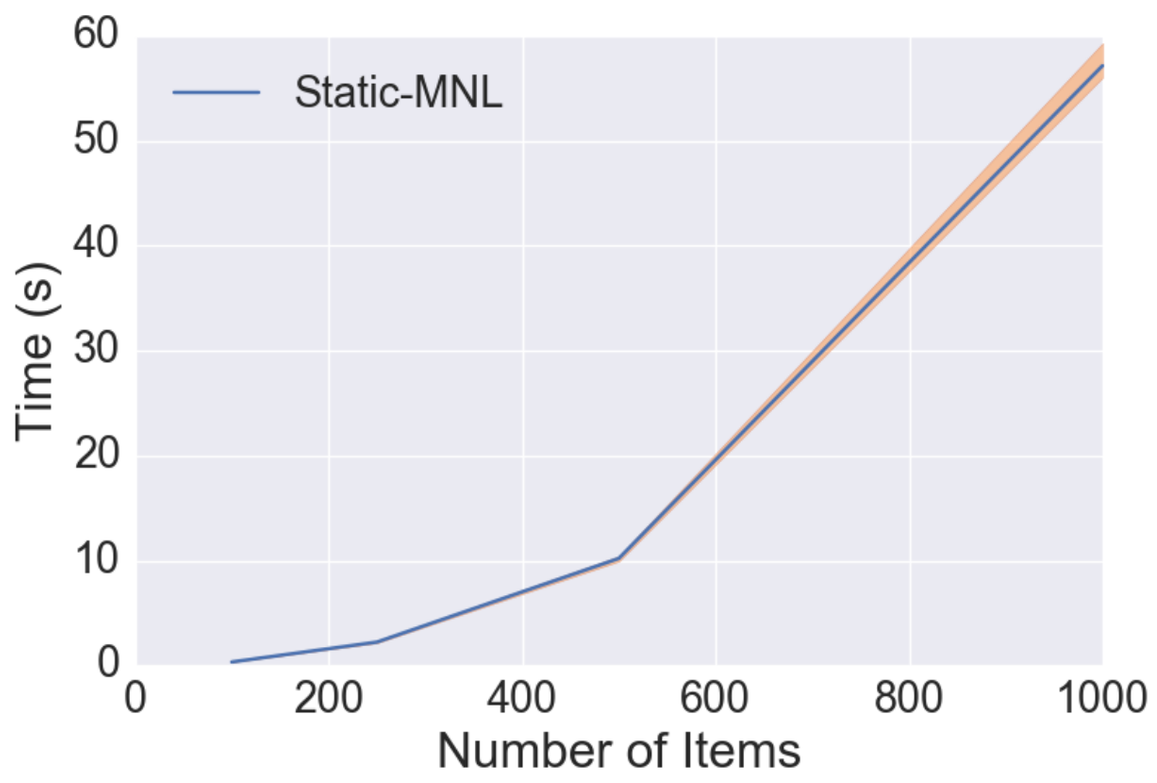


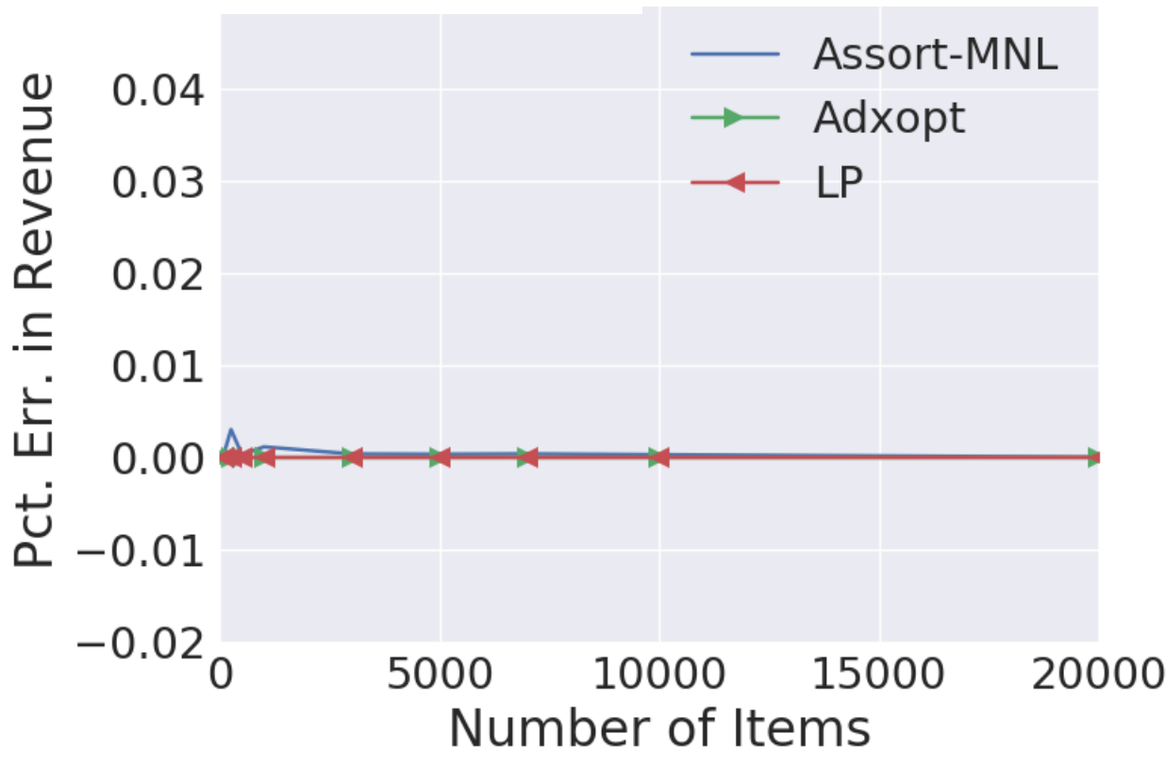
Experiments: General assortments



Experiments: Cardinality Constrained







Summary

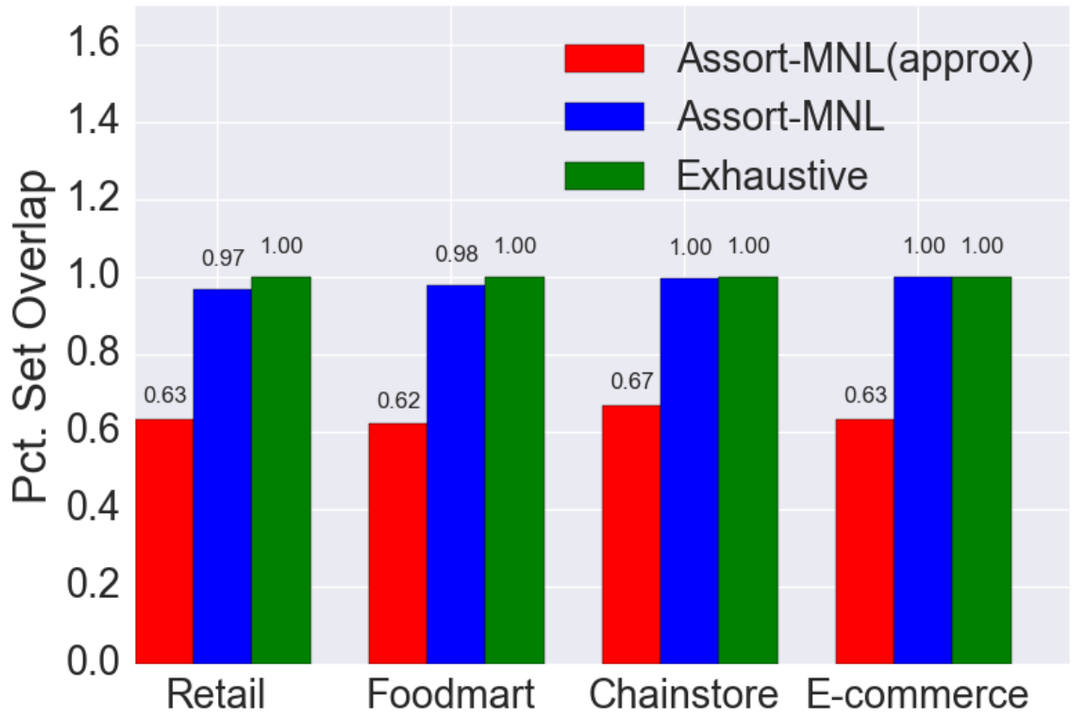
- Propose efficient algorithm for AO under MNL model with arbitrary feasible assortments
- Iterative algorithm that builds on binary search and fast methods for maximum inner product search
- Can trade-off accuracy and time
- For cardinality constrained AO, time complexity is linear in the number of products
- Algorithm relevant in both online and offline settings:
  - Online: Assort-MNL scales well as problem instances grow
  - Offline: Optimization over arbitrary assortments driven by business insights or transaction log mining

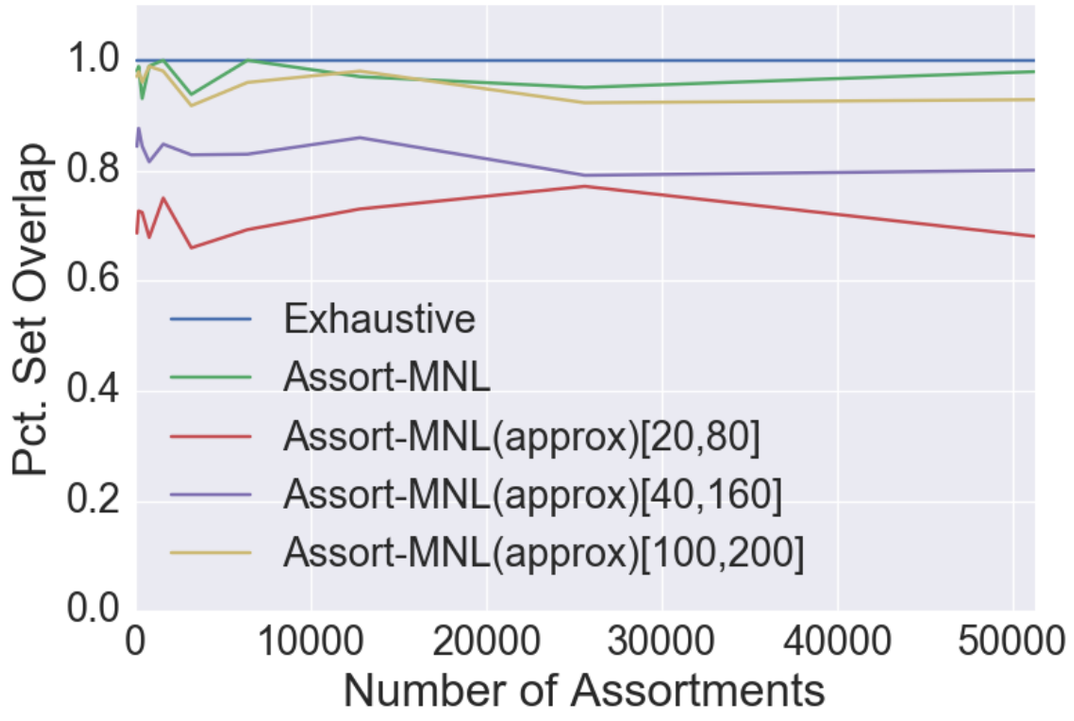
Future Directions

- Application of this method in other assortment planning problems such as robust assortment optimization
- With arbitrary assortments based on frequent itemset mining, model purchase of multiple items and perform AO

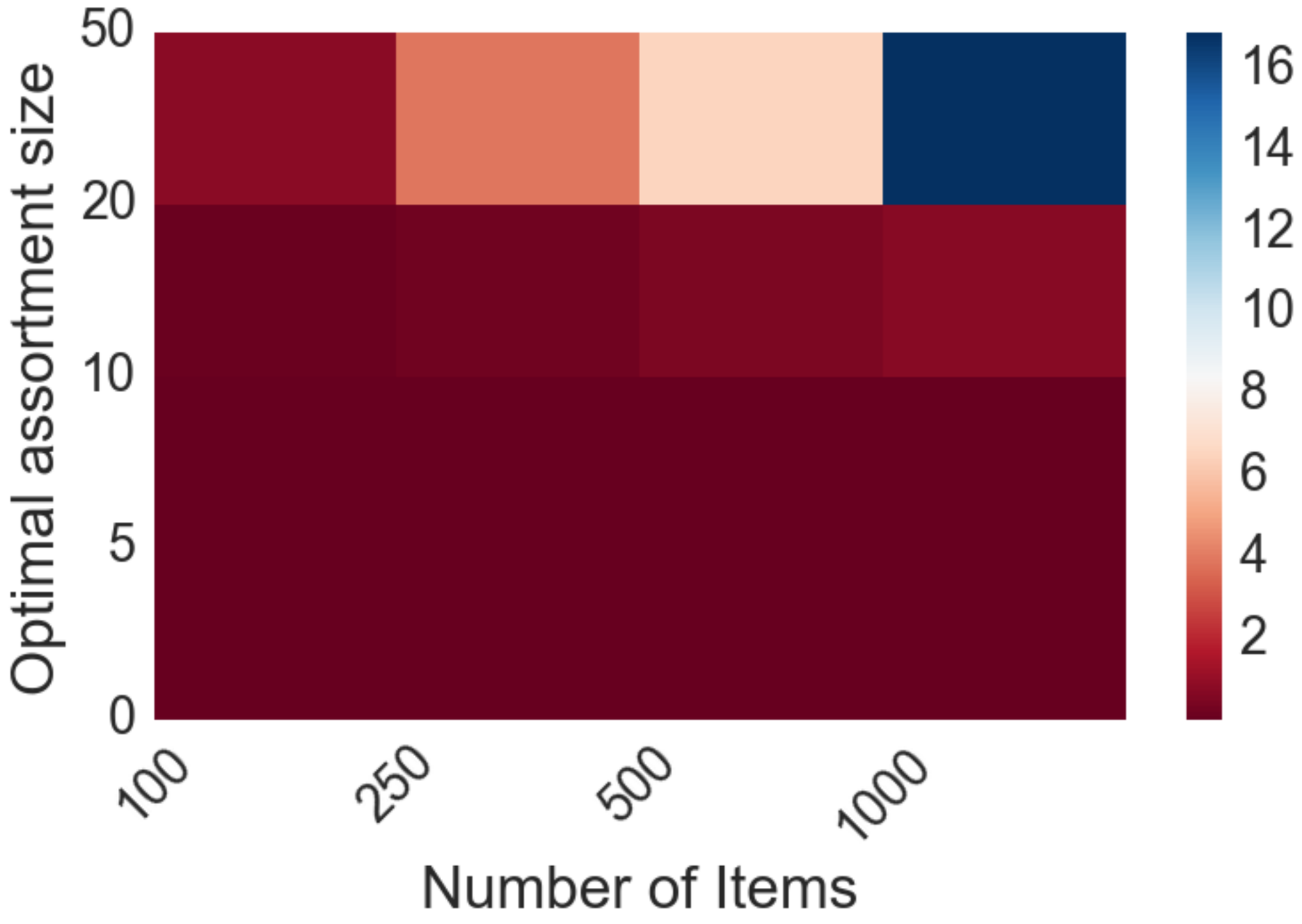
Thanks!

Experiments: General assortments





Experiments: General assortments



Assort-MNL(BZ): Guarantees

Theorem [S., Tulabandhula]: Under mild assumptions, Assort-MNL(BZ) algorithm leads to $P(|R^* - R| > \delta) \leq \theta$ with time complexity

$$O\left(nN^{\rho}\frac{\sqrt{\alpha}+0.5}{\log\frac{\theta\delta}{p_1-\delta}}\right)$$

where R^* is the revenue of the optimal assortment, α is the failure probability of the LSH structure, δ is the tolerance level and θ is the confidence level.

- Time taken is sub-linear in the number of assortments